Cells as Machines: Towards Deciphering Biochemical Programs in the Cell

François Fages

Cells as Machines: Towards Deciphering Biochemical Programs in the Cell

François Fages

Inria Paris-Rocquencourt, EPI Contraintes, France

Abstract. Systems biology aims at understanding complex biological processes in terms of their basic mechanisms at the molecular level in cells. The bet of applying theoretical computer science concepts and software engineering methods to the analysis of distributed biochemical reaction systems in the cell, designed by natural evolution, has led to interesting challenges in computer science, and new model-based insights in biology. In this paper, we review the development over the last decade of the biochemical abstract machine (Biocham) software environment for modeling cell biology molecular reaction systems, reasoning about them at different levels of abstraction, formalizing biological behaviors in temporal logic with numerical constraints, and using them to infer non-measurable kinetic parameter values, evaluate robustness, decipher natural biochemical processes and implement new programs in synthetic biology.

1 Introduction

At the end of the 90s, with the end of the human genome project, research in bioinformatics started to evolve, passing from the analysis of the genomic sequence and structural biology problems, to the analysis of complex post-genomic interaction networks: expression of RNA and proteins, protein-protein interactions, transport, signal transduction, cell cycle, etc. Systems biology [31] is the name given to a new pluridisciplinary research field, involving biologists, computer scientists, mathematicians, physicists, to promote a change of focus towards system-level understanding of high-level functions of living organisms, from their biochemical bases at the molecular level. The main outcome of this effort has been the creation of, and easy access to,

- databases and ontologies of cell components [2];
- repositories of models of cell processes [11], through the definition of common exchange formats such as the Systems Biology Markup Language (SBML) [28,27];
- model editors [33,19] and simulation tools [24,37], making it possible to reproduce *in silico* analyses in articles, with models published as supplementary material;
- and the construction of a whole cell predictive computational model of the bacterium *Mycoplasma genitalium* including its 525 genes by Karr et al.[29].

Formal methods from theoretical computer science have been successfully applied in systems biology to master the complexity of biological networks and decipher biological processes, mostly at the molecular and cellular levels. The distinction between syntax and semantics is particularly fruitful for designing modeling languages and for reasoning about biological systems at different levels of abstraction. While interaction diagrams are the key for interacting with biologists, their transcription in formal graphs or formal languages compels the modeler to eliminate any ambiguity, and enables the use of a wide variety of structural or dynamic analysis tools. In these approaches, the mathematical formalisms of ordinary differential equations (ODE) and partial derivate equations (PDE) appear as low-level languages on top of which high-level languages can be designed to directly reflect the structure of the interactions, and apply novel static analysis methods.

The use of Petri nets to model chemical processes was proposed in [39] together with standard Petri net tools for static analyses. The notion of T-invariant is a key tool for analyzing extreme fluxes and optimizing metabolic networks [50], and provides a definition of modules in biochemical networks [21]. P-invariants provide structural conservation laws that can be directly used to eliminate variables in mathematical models based on ordinary differential equation models [47]. The notion of siphons and traps provide sufficient conditions for persistence and accumulation of molecular species in a network of reactions [1,36]. Petri nets have also been generalized to handle continuous dynamics [34,35,44] and to model gene regulatory networks [10]. The use of process calculi from concurrency theory was also proposed in [41] and inspired subsequent work in several directions including stochastic modeling [38,40], space and membrane dynamics [8], and molecular biology combinatorics [15].

In this paper, we review the development over the last decade of the biochemical abstract machine (Biocham, `http://contraintes.inria.fr/biocham`) software environment for modeling cell biology molecular reaction systems, reasoning about them at different levels of abstraction, formalizing biological behaviors in temporal logic with numerical constraints, and using them to infer non-measurable kinetic parameter values, evaluate robustness, decipher natural biochemical processes and design new biochemical programs in synthetic biology.

2 Biochemical Reaction Systems

Let $\mathcal{S}$ be a finite set of s molecular species. A reaction is a triple $(\boldsymbol{s}, \boldsymbol{s'}, f)$, noted $\boldsymbol{s} \xrightarrow{f} \boldsymbol{s'}$, where $\boldsymbol{s}, \boldsymbol{s'} : \mathcal{S} \to \mathbb{N}$ are multisets over $\mathcal{S}$ (stoichiometric coefficients), and $f : \mathbb{R}^s \to \mathbb{R}$ is a mathematical function over molecule quantities, called the rate function. Multisets are used for representing reactants and products in reactions, and a reaction is fundamentally a multiset rewriting rule. The chemical metaphor based on multiset rewriting has been proposed in computer science to program concurrent processes [4,5] and to reason about concurrent programs [7]. However in biochemistry, the reaction rates of the reactions may differ by several orders of magnitude, and it is crucial for many properties to consider the

continuous-time dynamics of the reactions. Each reaction is thus supposed to be given with a rate function.

A limited number of reaction schemas occurs in biochemical reaction networks. *Binding* reactions of the form

$$A, B \stackrel{kAB}{\longrightarrow} C$$

bind two molecular compounds together, such as the *complexation* of two proteins or complexes to form a bigger complex, or the binding of a promotion factor (resp. an inhibitor) on a gene to activate (resp. inhibit) its transcription. The mass action law kinetics used in that reaction states that the rate of the reaction is proportional to the number of its reactants. The rate constant k represents the affinity of the two molecules to bind together. The inverse unbinding reaction is of the form

$$C \stackrel{k.C}{\longrightarrow} A, B$$

with again a mass action law kinetics, where the rate constant characterizes the stability of the complex.

A molecular species like a protein can also be modified under the action of an enzyme, such as a kinase for a *phosphorylation* reaction, or a phosphatase for a dephosphorylation reaction. This is represented by a reaction of the form

$$A \stackrel{v.A/(k+A)}{\longrightarrow} B$$

with a Michaelis-Menten kinetics. That rate function for enzymatic reactions results in fact from the reduction of the three elementary reactions with mass action law kinetics,

$$A, E \underset{k_2 C}{\overset{k_1.A.E}{\rightleftarrows}} C \stackrel{k_3.C}{\longrightarrow} B, E$$

by quasi-steady state approximation [45]. The same reaction schema can also be used to model the active *transport* of a molecule A from one compartment, to another compartment where A is denoted by B.

Synthesis reaction, such as the synthesis of an RNA by a gene activated by its promotion factor, are of the form

$$A \stackrel{v.A^n/(k+A)^n}{\longrightarrow} A, B$$

with a Hill kinetics of order n. That rate function provides a sigmoidal response, i.e. a switch-like behavior to the synthesis process, and comes from the reduction of a system of n cooperative reactions.

Degradation reactions of the form

$$A \stackrel{k.A}{\longrightarrow} \_$$

have the empty multiset as product, and either a mass action law kinetics in the case of spontaneous degradation, or a Michaelis-Menten or Hill kinetics in the case of an active degradation process under the action of other molecules.

These formal systems of reactions can be interpreted at different level of abstraction in a hierarchy of semantics. The most concrete interpretation is provided by the *Chemical Master Equation* (CME), which defines the probability of being in a state $\boldsymbol{x}$ at time t as

$$\frac{d}{dt}p^{(t)}(\boldsymbol{x}) = \sum_{j:\boldsymbol{x}-\boldsymbol{r}_j\geq 0} f_j(\boldsymbol{x}-\boldsymbol{v}_j).p^{(t)}(\boldsymbol{x}-\boldsymbol{v}_j) - \sum_{j=1}^{n} f_j(\boldsymbol{x}-\boldsymbol{v}_j).p^{(t)}(x)$$

where $\boldsymbol{v}_j$ is the change vector $\boldsymbol{s'}_j - \boldsymbol{s}_j$ of reaction j and $f_j(\boldsymbol{x})$ is the propensity of reaction j in state $\boldsymbol{x}$ defined by the rate function.

The *differential semantics* of a reaction system is a deterministic interpretation, which describes the time evolution of the mean $E[X(t)]$ by an ODE. The ODE derives from the CME by a first-order approximation. We have

$$\frac{d}{dt}E[X(t)] = \sum_{\boldsymbol{x}} \frac{d}{dt}p^{(t)}(\boldsymbol{x}) = \sum_{j=1}^{n} \boldsymbol{v}_j.E[f(X(t))]$$

which gives, by first-order approximation of the Taylor series about the mean,

$$\frac{d}{dt}\boldsymbol{\mu} = \sum_{j=1}^{n} \boldsymbol{v}_j.f(\boldsymbol{\mu}).$$

Given initial concentrations for species, such an ODE can be simulated by standard numerical methods for stiff systems.

For instance, the ODE associated to the reaction system

$$S, E \stackrel{10.S.E}{\longrightarrow} C \stackrel{10.C}{\longrightarrow} P, E, \quad P \stackrel{P}{\longrightarrow} S$$

is $dS/dt = k3.P - k1.E.S$, $dE/dt = k2.C - k1.E.S$, $dC/dt = k1.E.S - k2.C$, $dP/dt = k2.C - k3.P$. Figure 1 shows the amplification of the input E in the output P, in a simulation of that ODE with initial concentration 10 for S and a cosine function of time for the input E.

The *stochastic semantics* of a reaction system is defined by a Continuous Time Markov Chain (CTMC) over integer numbers of molecules (discrete concentration levels). The rate functions of the reactions lead to state transition probabilities after normalization by the sum of the propensities of each reaction in each state. The Stochastic Simulation Algorithm of Gillespie [20] provides a simulation method which computes numerical traces, most often similar to the ODE simulation for large numbers of molecules, but may exhibit qualitatively different behaviors in the case of small numbers of molecules, for instance in the case of gene expression as a gene usually is in one single copy in a cell.

The abstraction of the stochastic semantics by simply forgetting the probabilities, gives the non-deterministic *Petri net semantics* of the reactions, where the discrete states define the number of tokens in each place, and the transitions consume the reactant tokens and produce the product tokens [39].

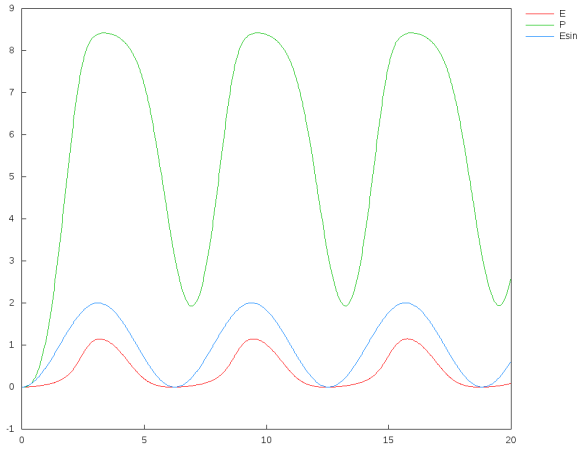

Fig. 1. Simulation of the time evolution of the concentration of output P in the differential semantics of the reaction system $S, E \overset{10.S.E}{\longrightarrow} C \overset{10.C}{\longrightarrow} P, E$, $P \overset{P}{\longrightarrow} S$, with initial concentration 10 for S, and a cosine function of time (depicted by E_{sin}) for input E.

The abstraction of the Petri net semantics in the *Boolean semantics* defined by the Boolean abstraction function over integers, $\beta : \mathbb{N} \longrightarrow \{0, 1\}$ with $\beta(0) = 0$ and $\beta(x) = 1$ if $x > 0$, is a non-deterministic asynchronous Boolean transition system suitable for reasoning on the presence/absence of molecules. In Biocham, the Boolean semantics of the reactions associates several Boolean transitions to one reaction. For instance, a complexation reaction like $A, B \longrightarrow B$, is interpreted by 4 Boolean transitions, one for each possible complete consumption of the 2 reactants: $A \wedge B \longrightarrow C \wedge \pm A \wedge \pm B$. This is necessary for the abstraction result to hold with respect to the Petri net or stochastic semantics. It is worth noticing that with a Boolean abstraction defined by a threshold value θ, i.e. $\beta_\theta(x) = 0$ if $x < \theta$ and $\beta_\theta(x) = 1$ if $x \geq \theta$, several Boolean transitions must be introduced for the products as well, for instance the complexation reaction gives rise to 16 Boolean transitions for taking into account the possible production of the 2 products, either below or above the threshold value.

In [18], all these discrete and stochastic trace semantics of reactions systems have been related by formal abstraction relationships (Galois connections) in the framework of abstract interpretation [14]. This shows that if a behavior is not possible in the Boolean semantics for instance, then it is not realizable in the Petri net or stochastic semantics for any kinetic laws and kinetic parameter values. This is a strong motivation for reasoning at a high level of abstraction in the Boolean semantics of reaction systems, which may be sufficient to answer questions about large interaction maps.

3 Symbolic Model-Checking of Biochemical Systems

Regulatory, signaling and metabolic networks are very complex mechanisms which are far from being understood on a global scale. Data on the kinetics of the individual reactions is also rare and unreliable, making the building of quantitative models particularly challenging in many cases. In those situations, qualitative analyses can however be conducted in the Boolean semantics of the reactions, using the powerful model-checking tools developed for circuit and program verification [13].

A Boolean state specifies the presence or absence of each molecule in the system at a given time, and any set of states can be represented by a Boolean constraint over the molecule variables. The *Computation Tree Logic* CTL* is a modal logic that extends propositional logic with two path quantifiers, **A** and **E** ($\mathbf{A}\phi$ meaning that ϕ is true on all computation paths, and $\mathbf{E}\phi$ that it is true on at least one path), and several temporal operators, $\mathbf{X}\phi$ (meaning that ϕ is true on the next state on a path), $\mathbf{F}\phi$ (meaning that ϕ is finally true on some state on a path), $\mathbf{G}\phi$ (globally true on all states on a path), $\phi\mathbf{U}\psi$ (until, meaning that ψ is finally true and ϕ is always true before), and $\phi\mathbf{R}\psi$ (release, meaning that ψ is either globally true or always true up to the first occurrence of ψ included). In this logic, $F\phi$ is equivalent to $\mathit{true}\mathbf{U}\phi$, $G\phi$ to $\phi\mathbf{R}\mathit{false}$, and we have the following duality properties: $\neg\mathbf{X}\phi = \mathbf{X}\neg\phi$, $\neg\mathbf{E}\phi = \mathbf{A}\neg\phi$, $\neg\mathbf{F}\phi = \mathbf{G}\neg\phi$, $\neg(\phi\mathbf{U}\psi) = \neg\psi\mathbf{R}\neg\phi$.

The fragment CTL of CTL* imposes that a temporal opertor must immediately follow a path quantifier. This logic CTL can express a wide variety of properties of biochemical networks [9] like state *reachability* of ϕ ($\mathbf{EF}\phi$), *steadyness* of ϕ ($\mathbf{EG}\phi$), *stability* ($\mathbf{AG}\phi$), reachability of a stable state ($\mathbf{EFAG}\phi$), ϕ *checkpoint* for ψ ($\neg\psi\mathbf{R}\phi$), *oscillations* ($\mathbf{EG}(\mathbf{F}\neg\phi \wedge \mathbf{F}\phi)$ over-approximated in CTL by $\mathbf{EG}(\mathbf{EF}\neg\phi \wedge \mathbf{EF}\phi)$) etc.

Figure 2 reproduces Kohn's map of the mammalian cell cycle [32] using some graphical conventions introduced by K. Kohn to represent the different types of interactions (complexation, binding, phosphorylations, modifications, synthesis, etc.). This map has been transcribed in a reaction model of 732 reaction rules over 165 proteins and genes, and 532 variables taking into account the different forms of the molecular species [9]. The astronomical number of Boolean states in this system, 2^{532}, prevents the explicit representation of the state graph, however, a set of states in this space can nevertheless be represented symbolically by a Boolean formula over 532 variables, and the transition relation by a Boolean formula over twice that number of variables. For instance the formula *false* represents the empty set, *true* the universe of all states, x the set of 2^{531} states where x is present, etc. Our first result in [9] was to show the performance of the state-of-the-art symbolic model checker NuSMV [12] using the representation of Boolean formulae by ordered binary decision diagrams (OBDD), on this non standard transition system from biology. Table 1 shows that the compilation of the whole 732 reactions into Boolean formulae took 29 seconds, and simple reachability and oscillations properties could be checked in a few seconds. The

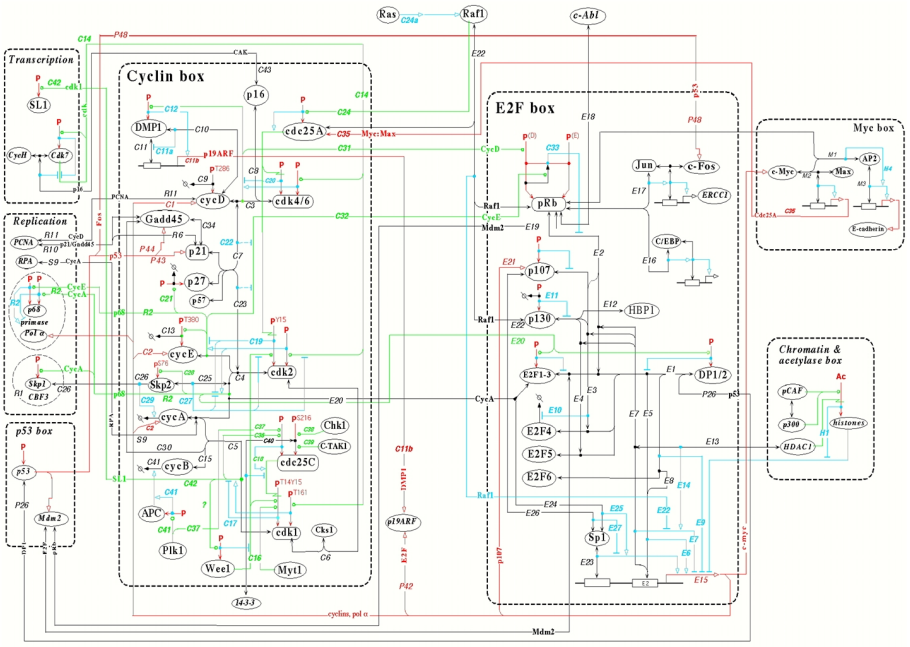

Fig. 2. Kohn's map of the mammalian cell cycle control [32].

negative answer to the query concerning the oscillation of cyclin B revealed the omission of the synthesis of cyclin B in Kohn's map.

The encoding of biological properties in temporal logics provides a *logical paradigm for systems biology* that makes a bridge between theoretical models and biological experiments, through the following identifications:

$$\begin{array}{c} \textit{biological model} = \textit{transition system},\\ \textit{biological property} = \textit{temporal logic formula},\\ \textit{model validation} = \textit{model-checking},\\ \textit{model inference} = \textit{constraint solving}. \end{array}$$

A formula ϕ, learned from biological experiments, can be tested in a model $\mathcal{M}$ by model-checking techniques to determine whether $\mathcal{M} \models \phi$. Furthermore, a model-checker can also compute the set of initial states for which a formula is true, and suggest biological experiments to verify a CTL property predicted by the model, on the real biological object [6]. In particular, the checkpoints proved in a model of the cell cycle, or of a signaling network, provide possible drug targets to block the cell cycle, or a signaling cascade.

CTL query	Answer	CPU time	whitness time
compilation of the reactions	-	29	-
reachable SL1(p)	yes	29	124
reachable cycE	yes	2	22
reachable cycD	yes	1.9	11.5
reachable pcna-cycD	yes	1.7	48.7
cdc25C(Nterm) checkpoint cdk1-cycB(Thr161))	no	2.2	49.22
oscillation cycA	yes	31.8	-
oscillation cycB	no	6	-

Table 1. Runtime in seconds obtained on Kohn's map with NuSMV in 2002 on a Pentium 3 at 600MHz, for checking simple CTL reachability and oscillation properties in a state corresponding to phase G2 of the cell cycle. The absence of possibility of oscillation for cycB corresponds to the omission of a reaction in Kohn's map, for the synthesis of cyclin B.

4 Quantitative Temporal Logic Constraints

4.1 Threshold and Timing Constraints

The temporal logic approach to the specification of imprecise dynamical properties of biological systems can also be made quantitative and applied to quantitative models over concentrations. The idea is to lift it to a first-order setting with numerical (linear) constraints over the reals, in order to express threshold or more complex constraints on the concentrations of the molecular compounds and time.

For instance, the reachability of a threshold concentration for a molecule A can be expressed with the formula $\mathbf{F}(A > v)$ for some value or free variable v. Such formulae can then be interpreted on a finite numerical trace (extended with a loop on the last state) obtained either from a biological experiment, or from the numerical simulation of an ODE model, giving the concentrations of the molecules at discrete time points, e.g. Figure3.

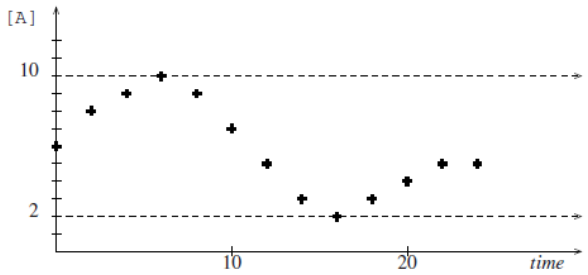

Fig. 3. Numerical trace depicting the time evolution of a protein concentration

In Biocham, we use the First-Order Linear Time Logic with linear constraints over the reals (FO-LTL($\mathbb{R}_{\text{lin}}$)) to specify semi-qualitative semi-quantitative properties of a biological dynamical system. LTL is the fragment of CTL* without any path quantifier and only time operators interpreted on a trace. The grammar of FO-LTL($\mathbb{R}_{\text{lin}}$) formulae is summarized in Table 2.

$$\phi ::= c \ \mid \phi \Rightarrow \psi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathbf{X}\phi \mid \mathbf{F}\phi \mid \mathbf{G}\phi \mid \phi \mathbf{U} \phi \ \mid \phi \mathbf{R} \phi$$

Table 2. *Grammar of FO-LTL($\mathbb{R}_{lin}$) formulae where c denotes linear constraints over molecular concentrations, their first derivative, free variables and the time variable.*

Timing constraints can be expressed with the time variable and free variables to relate the time of differents events. For instance, the formula $\mathbf{G}(Time \leq t_1 \Rightarrow [A] < 1 \wedge Time \geq t_2 \Rightarrow [A] > 10) \wedge (t_2 - t_1 < 60)$ expresses that the concentration of molecule A is always less than 1 up to some time t_1, always greater than 10 after time t_2, and the switching time between t_1 and t_2 is less than 60 units of time.

A local maximum for molecule concentration A can be defined with the formula $\mathbf{F}(A \leq x \wedge \mathbf{X}(A = x \wedge \mathbf{X}A \leq x))$. This formula can be used to define oscillation properties, with period constraints defined as time separation constraints between the local maxima of the molecule, as well as phase constraints between different molecules.

In [43,17], it is shown how the *validity domain* $\mathcal{D}_{(s_0,\ldots,s_n),\phi}$ of the free variables of an FO-LTL($\mathbb{R}_{\text{lin}}$) formula ϕ on a finite trace $(s_0, ..., s_n)$, can be computed by finite unions and intersections of polyhedra, by a simple extension of the model-checking algorithm, as follows:

- $\mathcal{D}_{(s_0,\ldots,s_n),\phi} = \mathcal{D}_{s_0,\phi}$,
- $\mathcal{D}_{s_i,c(\boldsymbol{x})} = \{\boldsymbol{v} \in \mathbb{R}^k \mid \ s_i \models c[\boldsymbol{v}/\boldsymbol{x}]\}$ for a constraint $c(\boldsymbol{x})$,
- $\mathcal{D}_{s_i,\phi\wedge\psi} = \mathcal{D}_{s_i,\phi} \cap \mathcal{D}_{s_i,\psi}$,
- $\mathcal{D}_{s_i,\phi\vee\psi} = \mathcal{D}_{s_i,\phi} \cup \mathcal{D}_{s_i,\psi}$,
- $\mathcal{D}_{s_i,\mathbf{X}\phi} = \mathcal{D}_{s_{i+1},\phi}$,
- $\mathcal{D}_{s_i,\mathbf{F}\phi} = \bigcup_{j=i}^{n} \mathcal{D}_{s_j,\phi}$,
- $\mathcal{D}_{s_i,\mathbf{G}\phi} = \bigcap_{j=i}^{n} \mathcal{D}_{s_j,\phi}$,
- $\mathcal{D}_{s_i,\phi\mathbf{U}\psi} = \bigcup_{j=i}^{n} (\mathcal{D}_{s_j,\psi} \cap \bigcap_{k=i}^{j-1} \mathcal{D}_{s_k,\phi})$.

For instance, on the numerical trace of Figure 3, the validity domain, depicted in Figure 4, of the formula $\mathbf{F}(A \geq y_1 \wedge \mathbf{F}(A \leq y_2))$, where y_1 and y_2 are free variables, is $y_1 \leq 10 \wedge y_2 \geq 2$.

4.2 Parameter Optimization

One major difficulty in quantitative systems biology, is that the kinetic parameter values of the biochemical reactions are usually unknown, and must be infered

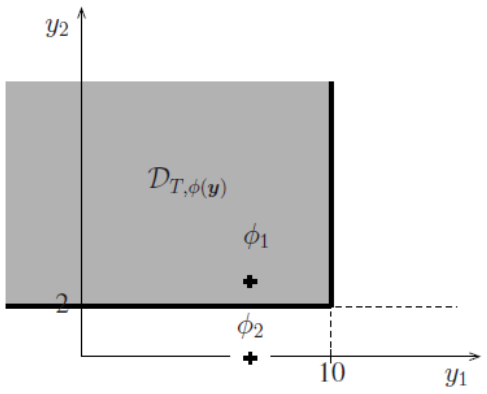

Fig. 4. Validity domain of the formula $\mathbf{F}(A \geq y_1 \wedge \mathbf{F}(A \leq y_2))$ on the trace of Figure 3. The two points correspond to the formulae $\phi_1 = \mathbf{F}(A \geq 7 \wedge \mathbf{F}(A \leq 3))$ (true) and $\phi_2 = \mathbf{F}(A \geq 7 \wedge \mathbf{F}(A \leq 0))$ (false) respectively.

from the observable behavior of the system under various conditions (differences of milieu, drugs, gene knock-outs or knock downs, etc.). In our quantitative temporal logic setting, this problem amounts to solve the inverse problem of finding parameter values for an ODE model such that an FO-LTL($\mathbb{R}_{\text{lin}}$) specification is true.

However, the classical true/false valuation of a logical formula is not well suited to guide the search. State-of-the-art continuous optimization algorithms such as evolutionary algorithms, require a fitness function to measure progress towards satisfiability. Such a continuous satisfaction degree in the interval $[0, 1]$ can be defined for FO-LTL($\mathbb{R}_{\text{lin}}$) formulae, by replacing constants by variables, which was in fact our original motivation for considering formulae with free variables.

Indeed, a specification of the expected behavior given by a closed formula, for instance

$$\phi_2 = \mathbf{F}(A \geq 7 \wedge \mathbf{F}(A \leq 0)),$$

can first be abstracted in a formula with free variables by replacing constants with free variables, e.g.

$$\phi = \mathbf{F}(A \geq y_1 \wedge \mathbf{F}(A \leq y_2))$$

with the objective values 7 for y_1 and 0 for y_2. Then, the validity domain $\mathcal{D}_{T,\phi}$ of the formula ϕ on a trace T obtained by simulation for some parameter values, makes it possible to define the *violation degree* $vd(T, \phi, o)$ of the formula on T with objective o, simply as the distance between the validity domain and the objective point o, i.e. 2 in our example (see Figure 4). A *continuous satisfaction degree* in the interval $[0, 1]$ can then be defined by normalization as the inverse

of the violation degree d plus one,

$$sd(T, \phi, o) = \frac{1}{1 + vd(T, \phi, o)}$$

i.e. 1/3 in our example.

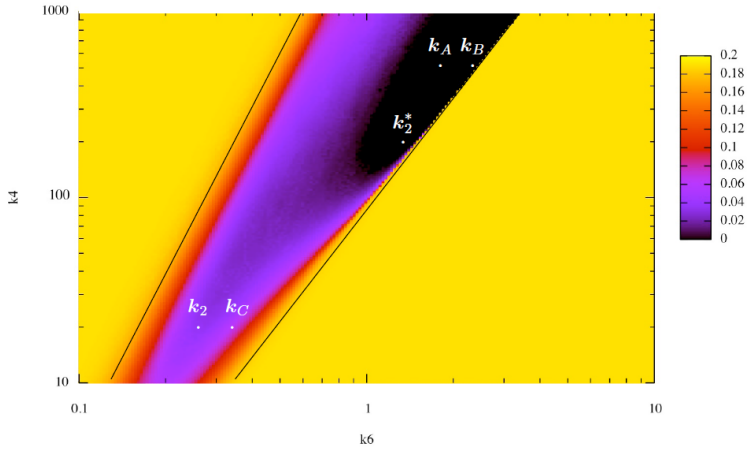

Fig. 5. Landscape of the satisfaction degree of an oscillation property with amplitude constraint, on a color scale from yellow to black, as a function of two parameters in a quantitative model of the yeast cell cycle from [48]. The parameter sets $\boldsymbol{k}_A$, $\boldsymbol{k}_B$ and $\boldsymbol{k}_2^*$ satisfy the specification. The parameter sets $\boldsymbol{k}_c$ and $\boldsymbol{k}_2$ violate the amplitude constraint. CMA-ES iteratively samples the landscape to find a path in a random walk from $\boldsymbol{k}_2$ to $\boldsymbol{k}_2^*$ for instance.

In Biocham, we use the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) of N. Hansen [22] as a black-box optimization algorithm, with the satisfaction degree of an FO-LTL($\mathbb{R}_{\text{lin}}$) specification as fitness function, and unknown kinetic parameter values (initial concentrations and control parameters) as variables. On a quantitative model of the cell cycle [48], Figure 5 depicts the landscape of the satisfaction degree of an oscillation property with amplitude constraint, as a function of two parameters of the model. The landscape is iteratively sampled by CMA-ES to find a path towards satisfaction, and optimize the model parameter values, for instance going from $\boldsymbol{k}_2$ to $\boldsymbol{k}_2^*$ in a few steps.

The FO-LTL($\mathbb{R}_{\text{lin}}$) satisfaction problem generalizes the classical curve fitting problem, by providing a powerful language to express significant properties of the dynamics, instead of requiring a complete curve that could over-specify the behavior. This is particularly useful in biology where experimental data may be imprecise in nature, with important cell-to-cell variability, irregular oscillation periods and phases, and should not be taken as exact specification.

This strategy for optimizing parameters with respect to an FO-LTL($\mathbb{R}_{\text{lin}}$) specification allowed us to solve a wide variety of problems in systems biology,

for fitting models to experimental data in high dimension (up to 100 parameters), revisiting the structure of the reaction network in case of failure, making new biological hypotheses based on simulation, and verifying them by new experiments, for instance for deciphering the complex dynamics of a cell signaling network in [23]. The same strategy for parameter optimization can also be used to compute control parameters to achieve a desired behavior at the single cell of cell population levels. This has been used for the model-based real-time control of gene expression in yeast cells using a microfluidic device in [49], and at the whole body scale, to couple models of cell cycle, circadian clock, drug effects, DNA repair system, and optimize anti-cancer drug chronotherapeutics in [16,3].

4.3 Robustness Measure

In [30], Kitano gives a general definition of the robustness of a property ϕ of a system S with respect to a set P of perturbations given with their probability distribution, as the mean functionality of the system with respect to ϕ under the perturbations, with the system's functionality defined in an *ad hoc* way for each property.

In our framework, this definition can be instanciated to a complete definition for FO-LTL($\mathbb{R}_{\text{lin}}$) properties, simply by taking their continuous satisfaction degree as functionality measure, as follows [42]:

$$\mathcal{R}_{S,\phi,P} = \int_{p \in P} prob(p)\ sd(T_p, \phi)\ dp.$$

In a model, this definition of robustness can be evaluated by

1. sampling the perturbations according to their distribution;
2. measuring the satisfaction degree of the property for each simulation of the perturbed model;
3. and returning the average satisfaction degree.

This methodology has been used in [42] to design and implement in synthetic biology using a cascade of gene inhibitions, a robust switch satisfying some timing constraints. Moreover, continuous parameter sensitivity indices can be computed in this approach to determine the most important parameters for improving the robustness of the design.

On the quantitative model of the yeast cell cycle [48] and the oscillation with amplitude constraint depicted in Figure 5, the estimated degree of robustness for parameters $\boldsymbol{k}_A$, $\boldsymbol{k}_B$ and $\boldsymbol{k}_C$ are respectively 0.991, 0.917 and 0.932. This is consistent with the location of points $\boldsymbol{k}_A$, $\boldsymbol{k}_B$ and $\boldsymbol{k}_C$. Perturbations around point $\boldsymbol{k}_A$ have high probabilities of staying in the region satisfying the specification whereas perturbations around point $\boldsymbol{k}_B$ have high probabilities of moving the system to the region with no oscillation. $\boldsymbol{k}_C$ is more robust than $\boldsymbol{k}_B$ even though, as opposed to $\boldsymbol{k}_B$, its violation degree is non null. This is explained by the abrupt transition between oscillating and non oscillating regions near $\boldsymbol{k}_B$ compared to the smoother transition near $\boldsymbol{k}_C$.

5 Biochemical Programming

Synthetic biology prolongs systems biology with the aim of designing biological systems that perform novel, useful tasks, and implementing them *in vivo* by reengineering and optimizing existing natural organisms. This is achieved by modifying the genes or integrating DNA constructs in living cells, or by creating cell-free vesicles, using bioengineering techniques. Synthetic biology keeps modeling and the characterization of components as central methodology to achieve its goals. Some successes of this nascent field include: the constitution of registries of standard biological parts and the organization of the iGEM competition at MIT; the creation by Craig Venter of a cell with a synthetic genome; the production by Sanofi of artemisinin, an antimalarial drug, by a biosynthetic pathway in a yeast chassis.

However, in order to design robust interaction networks and to be reliable in a clinical context, synthetic circuits must progress in their biochemical implementation of logical tasks and simple operations.

One way to attack this problem is to study the compilation of imperative programs in biochemical reaction systems over proteins. In [46], Senum and Riedel have shown how Boolean and arithmetic operations can be robustly implemented with biochemical reactions using mass action law kinetics, and only two kinetic rate constants s and f, for fast and slow reactions respectively. These transformations use an intermediate language of *conditional reactions* with preconditions. The preconditions are logical expressions over Boolean variables associated to each molecular species. The Boolean truth values are defined from the concentrations with a threshold function β_θ as in Section 2.

For instance, a reaction with precondition A is simply transformed by adding A as catalyst (i.e. both reactant and product). For a disjunctive precondition, $A \vee B$, two reactions are created, one with A and one with B as catalyst. A negation in a precondition amounts to test the absence of a molecular species which cannot be directly done in a biochemical reaction. The idea is to introduce a witness molecule A' for the absence of A without affecting A, using the following slow and fast mass action law kinetic reactions: $\_ \stackrel{s}{\longrightarrow} A'$, $A, A' \stackrel{f.A.A'}{\longrightarrow} A$, $2{*}A' \stackrel{f.A'^2}{\longrightarrow} A$.

For the copy instruction, `B:=A`, compiling it with just one reaction $A \longrightarrow B$ would destroy A. On the other hand, the reaction $A \longrightarrow A, B$ would increase B at each increment of A. In order to localize the computation for the copy, the following conditional reactions are used

$$\begin{array}{ll} A \longrightarrow C & \text{precondition } G \\ G \longrightarrow \_ & \text{precondition } \neg A \\ C \longrightarrow A, B & \text{precondition } \neg G \end{array}$$

where G is a start signal molecule for executing the instruction and which is consumed in the process. This is the basic idea to implement arithmetic operations and comparisons thourhg asynchronuous biochemical computation.

In [25], the authors further extend this approach to the compilation of program control flows. For instance, the following program for the Euclidean division

of A by B, is compiled, first in a conditional reaction program where initially Q is zero and C is initially of a unit amount:

`Q:=0`	$A, B \longrightarrow D$
`while A>=B do`	$C \longrightarrow Q, E$ precondition $\neg B$
`begin`	$D \longrightarrow F$ precondition $\neg C$
`  A:=A-B;`	$E \longrightarrow G$ precondition $\neg D$
`  Q:=Q+1;`	$F \longrightarrow B$ precondition $\neg E$
`end;`	$G \longrightarrow C$ precondition $\neg F$
`R:=A`	$D \longrightarrow R$ precondition $B \wedge \neg A$

and then into a system of biochemical reactions with only two fast and slow mass action law kinetics. The execution with initial concentrations $[A] = 20$ and $[B] = 3$ produces the result $[Q] = 6,\ [R] = 2$ as follows:

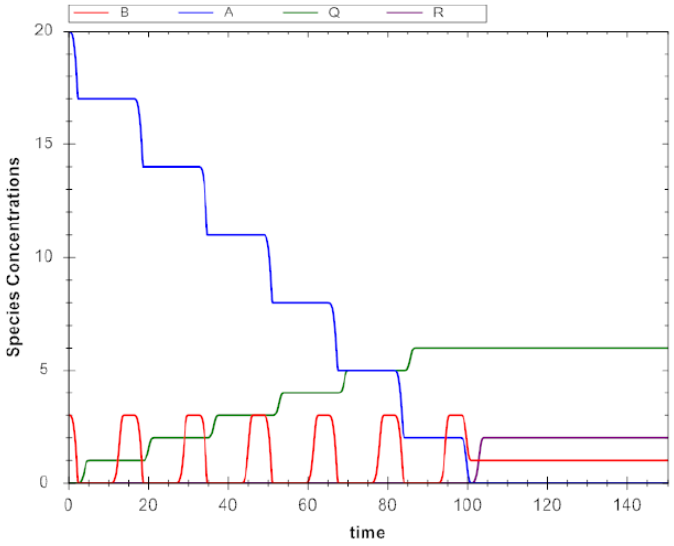

Fig. 6. Biochemical computation of the Euclidean division of A by B [25].

However, more work is needed on this schema to minimize the number of involved molecular species [26]. This is crucial to accomplish a complex computation within a confined biochemical environment. The challenge of implementing simple imperative programs with protein reaction systems in vesicles seems atteinable in a near future with enormous applications for creating biosensors and personalized therapeutics at the microscopic scale.

6 Conclusion

This line of research in systems biology based on the vision of cell as computation, aims at mastering the complexity of cell processes, through the use of concepts and tools from theoretical computer science and the establishment of formal computation paradigms tightly coupled to experimental settings in cell biology. While for the biologist, as well as for the mathematician, the sizes of the biological networks and the number of elementary interactions constitute a complexity barrier, for the computer scientist the difficulty is not that much in the size of the networks than in the unconventional nature of biochemical computation. Unlike most programs, biochemical computation involve transitions that are stochastic rather than deterministic, continuous-time rather than discrete-time, poorly localized in compartments instead of well-structured in modules, and created by evolution instead of by rational design. It is our belief however that some form of modularity (functional if not structural) is required by an evolutionary system to survive, and that the elucidation of these modules in biochemical computation is now a key to master the analog aspects of biochemical computation, understand natural biochemical programs, and start controlling the cell machinery.

References

1. D. Angeli, P. De Leenheer, and E. D. Sontag. A petri net approach to persistence analysis in chemical reaction networks. In *Biology and Control Theory: Current Challenges*, volume 357 of *LNCIS*, pages 181–216. Springer-Verlag, 2007.
2. Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.
3. A. Ballesta, S. Dulong, C. Abbara, B. Cohen, A. Okyar, J. Clairambault, and F. Levi. A combined experimental and mathematical approach for molecular-based optimization of irinotecan circadian delivery. *PLOS Computational Biology*, 7(9), 2011.
4. Jean-PIerre Banâtre and Daniel Le Métayer. Chemical reaction as a computational model. *Functional Programmming*, pages 103–117, 1989.
5. Jean-PIerre Banâtre and Thierry Priol. Chemical programming of future service-oriented architectures. *Jounral of Software*, 4:738–746, September 2009.
6. Gilles Bernot, Jean-Paul Comet, Adrien Richard, and J. Guespin. A fruitful application of formal methods to biological regulatory networks: Extending thomas' asynchronous logical approach with temporal logic. *Journal of Theoretical Biology*, 229(3):339–347, 2004.
7. Gérard Berry and Gérard Boudol. The chemical abstract machine. *Theoretical Computer Science*, 96, 1992.
8. Luca Cardelli. Brane calculi - interactions of biological membranes. In Vincent Danos and Vincent Schächter, editors, *CMSB'04: Proceedings of the second international workshop on Computational Methods in Systems Biology*, volume 3082 of *Lecture Notes in BioInformatics*, pages 257–280. Springer-Verlag, 2004.

9. Nathalie Chabrier-Rivier, Marc Chiaverini, Vincent Danos, François Fages, and Vincent Schächter. Modeling and querying biochemical interaction networks. *Theoretical Computer Science*, 325(1):25–44, September 2004.
10. Claudine Chaouiya. Petri net modelling of biological networks. *Briefings in Bioinformatics*, 2007.
11. Vijayalakshmi Chelliah, Camille Laibe, and Nicolas Novère. Biomodels database: A repository of mathematical models of biological processes. In Maria Victoria Schneider, editor, *In Silico Systems Biology*, volume 1021 of *Methods in Molecular Biology*, pages 189–199. Humana Press, 2013.
12. Alessandro Cimatti, Edmund Clarke, Fausto Giunchiglia Enrico Giunchiglia, Marco Pistore, Marco Roveri, Roberto Sebastiani, and Armando Tacchella. Nusmv 2: An opensource tool for symbolic model checking. In *Proceedings of the International Conference on Computer-Aided Verification, CAV'02*, Copenhagen, Danmark, July 2002.
13. Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking*. MIT Press, 1999.
14. Patrick Cousot and Radhia Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL'77: Proceedings of the 6th ACM Symposium on Principles of Programming Languages*, pages 238–252, New York, 1977. ACM Press. Los Angeles.
15. Vincent Danos and Cosimo Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69–110, 2004.
16. Elisabetta De Maria, François Fages, Aurélien Rizk, and Sylvain Soliman. Design, optimization, and predictions of a coupled model of the cell cycle, circadian clock, dna repair system, irinotecan metabolism and exposure control under temporal logic constraints. *Theoretical Computer Science*, 412(21):2108–2127, May 2011.
17. François Fages and Aurélien Rizk. On temporal logic constraint solving for the analysis of numerical data time series. *Theoretical Computer Science*, 408(1):55–65, November 2008.
18. François Fages and Sylvain Soliman. Abstract interpretation and types for systems biology. *Theoretical Computer Science*, 403(1):52–70, 2008.
19. Akira Funahashi, Yukiko Matsuoka, Akiya Jouraku, Mineo Morohashi, Norihiro Kikuchi, and Hiroaki Kitano. Celldesigner 3.5: A versatile modeling tool for biochemical networks. *Proceedings of the IEEE*, 96(8):1254–1265, August 2008.
20. Daniel T. Gillespie. General method for numerically simulating stochastic time evolution of coupled chemical-reactions. *Journal of Computational Physics*, 22:403–434, 1976.
21. Eva Grafahrend-Belau, Falk Schreiber, Monika Heiner, Andrea Sackmann, Björn H. Junker, Stefanie Grunwald, Astrid Speer, Katja Winder, and Ina Koch. Modularization of biochemical networks based on a classification of petri net by T-invariants. *BMC Bioinformatics*, 9(90), February 2008.
22. Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
23. Domitille Heitzler, Guillaume Durand, Nathalie Gallay, Aurélien Rizk, Seungkirl Ahn, Jihee Kim, Jonathan D. Violin, Laurence Dupuy, Christophe Gauthier, Vincent Piketty, Pascale Crépieux, Anne Poupon, Frédérique Clément, François Fages, Robert J. Lefkowitz, and Eric Reiter. Competing g protein-coupled receptor kinases balance g protein and β-arrestin signaling. *Molecular Systems Biology*, 8(590), June 2012.

24. Stefan Hoops, Sven Sahle, Ralph Gauges, Christine Lee, Jürgen Pahle, Natalia Simus, Mudita Singhal, Liang Xu, Pedro Mendes, and Ursula Kummer. Copasi – a complex pathway simulator. *Bioinformatics*, 22(24):3067–3074, 2006.
25. De-An Huang, Jie-Hong Jiang, Ruei-Yang Huang, and Chi-Yun Cheng. Compiling program control flows into biochemical reactions. In *ICCAD'12: IEEE/ACM International Conference on Computer-Aided Design*, San Jose, USA, November 2012.
26. Ruei-Yang Huang, De-An Huang, Hui-Ju Katherine Chiang, Jie-Hong Jiang, and François Fages. Species minimization in computation with biochemical reactions. In *IWBDA'13: Proceedings of the fifth International Workshop on Bio-Design Automation*, Imperial College, London, July 2013.
27. Michael Hucka et al. The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.
28. Michael Hucka, Stefan Hoops, Sarah M. Keating, Le Novère Nicolas, Sven Sahle, and Darren Wilkinson. Systems biology markup language (SBML) level 2: Structures and facilities for model definitions. *Nature Precedings*, December 2008.
29. Jonathan R. Karr, Jayodita C. Sanghvi, Derek N. Macklin, Miriam V. Gutschow, Jared M. Jacobs, Benjamin Bolival Jr, Nacyra Assad-Garcia, John I. Glass, , and Markus W. Covert. A whole-cell computational model predicts phenotype from genotype. *Cell*, 150(2):389,401, 2012.
30. H. Kitano. Towards a theory of biological robustness. *Molecular Systems Biology*, 3:137, 2007.
31. Hiroaki Kitano. Systems biology: A brief overview. *Science*, 295(5560):1662–1664, March 2002.
32. Kurt W. Kohn. Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Molecular Biology of the Cell*, 10(8):2703–2734, August 1999.
33. Nicolas le Novere, Michael Hucka, Huaiyu Mi, Stuart Moodie, Falk Schreiber, Anatoly Sorokin, Emek Demir, Katja Wegner, Mirit I. Aladjem, Sarala M. Wimalaratne, Frank T. Bergman, Ralph Gauges, Peter Ghazal, Hideya Kawaji, Lu Li, Yukiko Matsuoka, Alice Villeger, Sarah E. Boyd, Laurence Calzone, Melanie Courtot, Ugur Dogrusoz, Tom C. Freeman, Akira Funahashi, Samik Ghosh, Akiya Jouraku, Sohyoung Kim, Fedor Kolpakov, Augustin Luna, Sven Sahle, Esther Schmidt, Steven Watterson, Guanming Wu, Igor Goryanin, Douglas B. Kell, Chris Sander, Herbert Sauro, Jacky L. Snoep, Kurt Kohn, and Hiroaki Kitano. The systems biology graphical notation. *Nature Biotechnology*, 27(8):735–741, August 2009.
34. Hiroshi Matsuno, Atsushi Doi, Masao Nagasaki, and Satoru Miyano. Hybrid petri net representation of gene regulatory network. In *Proceedings of the 5th Pacific Symposium on Biocomputing*, pages 338–349, 2000.
35. Hiroshi Matsuno, Yukiko Tanaka, Hitoshi Aoshima, Atsushi Doi, Mika Matsui, and Satoru Miyano. Biopathways representation and simulation on hybrid functional petri net. *In Silico Biology*, 3:32, 2003.
36. Faten Nabli, François Fages, Thierry Martinez, and Sylvain Soliman. A boolean model for enumerating minimal siphons and traps in petri-nets. In *Proceedings of CP'2012, 18th International Conference on Principles and Practice of Constraint Programming*, volume 7514 of *Lecture Notes in Computer Science*, pages 798–814. Springer-Verlag, October 2012.

37. Masao Nagasaki, Shuichi Onami, Satoru Miyano, and Hiroaki Kitano. Bio-calculus: Its concept, and an application for molecular interaction. In *Currents in Computational Molecular Biology*, volume 30 of *Frontiers Science Series*. Universal Academy Press, Inc, 2000. This book is a collection of poster papers presented at the RECOMB 2000 Poster Session.
38. C. Priami, A. Regev, W. Silverman, and E. Shapiro. Application of a stochastic name passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 80:25–31, 2001.
39. Venkatramana N. Reddy, Michael L. Mavrovouniotis, and Michael N. Liebman. Petri net representations in metabolic pathways. In Lawrence Hunter, David B. Searls, and Jude W. Shavlik, editors, *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 328–336. AAAI Press, 1993.
40. Aviv Regev, Ekaterina M. Panina, William Silverman, Luca Cardelli, and Ehud Shapiro. Bioambients: An abstraction for biological compartments. *Theoretical Computer Science*, 325(1):141–167, September 2004.
41. Aviv Regev, William Silverman, and Ehud Y. Shapiro. Representation and simulation of biochemical processes using the pi-calculus process algebra. In *Proceedings of the sixth Pacific Symposium of Biocomputing*, pages 459–470, 2001.
42. Aurélien Rizk, Grégory Batt, François Fages, and Sylvain Soliman. A general computational method for robustness analysis with applications to synthetic gene networks. *Bioinformatics*, 12(25):il69–il78, June 2009.
43. Aurélien Rizk, Grégory Batt, François Fages, and Sylvain Soliman. Continuous valuations of temporal logic specifications with applications to parameter optimization and robustness measures. *Theoretical Computer Science*, 412(26):2827–2839, 2011.
44. Christian Rohr, Wolfgang Marwan, and Monika Heiner. Snoopy - a unifying petri net framework to investigate biomolecular networks. *Bioinformatics*, 26(7):974–975, 2010.
45. Lee A. Segel. *Modeling dynamic phenomena in molecular and cellular biology*. Cambridge University Press, 1984.
46. Philipp Senum and Marc Riedel. Rate-independent constructs for chemical computation. *PLOS One*, 6(6), 2011.
47. Sylvain Soliman. Invariants and other structural properties of biochemical models as a constraint satisfaction problem. *Algorithms for Molecular Biology*, 7(15), May 2012.
48. John J. Tyson. Modeling the cell division cycle: cdc2 and cyclin interactions. *Proceedings of the National Academy of Sciences*, 88(16):7328–7332, August 1991.
49. Jannis Uhlendorf, Agnés Miermont, Thierry Delaveau, Gilles Charvin, François Fages, Samuel Bottani, Gregory Batt, and Pascal Hersen. Long-term model predictive control of gene expression at the population and single-cell levels. *Proceedings of the National Academy of Sciences USA*, 109(35):14271–14276, 2012.
50. Ionela Zevedei-Oancea and Stefan Schuster. Topological analysis of metabolic networks based on petri net theory. *In Silico Biology*, 3(29), 2003.